International Journal of Research in Engineering, Science and Management
Volume 4, Issue 6, June 2021
https://www.ijresm.com | ISSN (Online): 2581-5792

A Modified 1D-CNN Based Network Intrusion Detection System

Anju Krishnan[1*], S. T. Mithra[2]
[1]*Student, Department of Electronics and Communication Engineering, Sree Buddha College of Engineering, Pathanamthitta, India*
[2]*Assistant Professor, Department of Electronics and Communication Engineering, Sree Buddha College of Engineering, Pathanamthitta, India*

***Abstract*: One of the crucial components of network security is intrusion detection. The well-received detection technology used conventional machine learning techniques to train the intrusion samples. Since the accuracy of intrusion detection is not commendable in traditional ML technologies, in this paper, we investigate and research a deep learning approach for developing a versatile IDS using a one-dimensional Convolutional Neural Network (1DCNN) which is normally used for supervised learning on time-series data. In each experiment, the CNN models are raced up to 10, 20, 30, and 40 epochs. For comparing the performance along with 1D-CNN, SVM and Naïve Bayes techniques are also utilized. 1D-CNN have outperformed compared to other two techniques. This is mainly because of the rationale that CNN has the potential to extract high-level feature representations.**

***Keywords*: 1D-CNN, Deep Learning, IDS, Machine Learning, Naïve Bayes, NSL-KDD, SVM.**

1. Introduction

To identify abnormal behaviour or attacks in a network IDS are being widely used. Intrusion detection is said to be efficient if False alarm rate is less and detection rate is commendable. In this paper, deep learning-based intrusion detection is introduced, and the experimental results shows that the proposed model helps to identify cyberattacks effectively. This paper relies on one-dimensional convolutional neural networks (1D-CNN) based IDS.

In brief, we use convolutional layers with RELU activation functions followed by pooling layers and fully connected layers, and finally soft-max function for classification of samples. Our proposed framework is shown in Fig. 1. We depict the usability of 1DCNN based IDS by applying it on NSL-KDD knowledge dataset and compare it with Naïve Bayes and SVM ML techniques. In the following sections, Literature review of the reference papers are provided in section 2. Our proposed system and its framework along with a block diagram are depicted in Section 3. Along with the explanation of output shapes of CNN model, a comparison of the performance with Naïve Bayes and SVM models are detailed in Section 4.

2. Literature Review

Riaz Ullah Khan, Xiaosong Zhang, Mamoun Alazab and Rajesh kumar [1] proposes a CNN based IDS. For compasison of performance SVM, DBN, and CNN algorithms are exploited by using KDD99 Dataset. The detection effect of the CNN system is detected to be higher than that of other algorithms. Meliboev Azizjon, Alikhanov Jumabek and Wooseong Kim [2] developed a flexible IDS using a 1D Convolutional Neural Network. The model is evaluated using the UNSW-NB15 IDS dataset and compared with Random Forest (RF) and Support Vector Machine (SVM). Anish Halimaa A and Dr. K. Sundarakantham [3] investigated and compared machine Learning technologies like SVM and Naïve Bayes that are well-known to solve the classification problems and NSL– KDD Dataset is utilized for performance evaluation. The outcomes show that SVM classifies better than Naïve Bayes.

Setareh Roshan [4] addresses the problem of adaptability in the field of intrusion detection by proposing a new intrusion detection system. Two novel approaches, Extreme learning machines (ELM) and Online sequential learning (OS-ELM) were presented. To evaluate the performance of the proposed IDS, the experiments were done in two modes of supervised and unsupervised using the NSL-KDD data set. The experiments showed that the system was able to detect both known and novel traffics while providing better rates of detection and false positives. system. I. Ahmad [5] investigated and compared SVM, Random Forest, and ELM using NSL–KDD dataset. For half and 1/4 of the data samples, SVM shows better results. Testing on the full data samples, ELM gives more accurate output. Buse Gul Atli [6] developed an IDS that combines ELM and statistical measurements. This paper explains the efficiency of the proposed method and provides a performance evaluation using the ISCX-IDS 2012 dataset. Since feature selection methodology is not used time consumed for the learning process is high.

Mehrnaz Mazini [8] explained artificial bee colony (ABC) and AdaBoost algorithms. These two are A-IDS methods. ABC algorithm is used to feature selection and AdaBoost is used to evaluate and classify the features. Results of the evaluation on

*Corresponding author: anju90k@gmail.com

NSL-KDD and ISCXIDS 2012 datasets confirm that this method has a very visible difference from other IDS. Wathiq Laftah Al-Yaseen, Zulaiha Ali Othman, and Mohd Zakree Ahmad Nazri [9] describe a multi-level IDS model that combines SVM with ELM and for evaluation purpose the KDD Cup 1999 dataset is used. It requires less training time and gives more accuracy. In this proposed model, more than one classifier is used, and hence longer testing time is required when compared with methods using only one classifier. Vajiheh Hajisalem and Shahram Babaie [10] introduce a model using Artificial Bee Colony (ABC) and Artificial Fish Swarm (AFS) algorithms along with the Fuzzy C-Means Clustering (FCM) and Correlation-based Feature Selection (CFS). The simulation results on NSL-KDD and UNSW-NB15 can achieve a 99% detection rate and 0.01% false-positive rate. Mohammed A. Ambusaidi and Priyadarsi Nanda [11] discovered a mutual information-based algorithm i.e., LSSVM-IDS using 3 datasets, which are KDD Cup 99, NSL-KDD, and Kyoto 2006+ dataset. The evaluation results show better accuracy and lower computational. Seyed Reza Hasani, Zulaiha Ali Othman and Seyed Mostafa Mousavi Kahaki [12] proposed Linear Genetic Programming (LGP) with Bees Algorithm which helps to improve the accuracy and efficiency of the proposed System.

3. Proposed Model

Fig. 1 is the block diagram of the proposed model used in this paper. It can be seen in Fig. 1, that the framework mainly consists of three steps:

1. Data preprocessing
2. Training and Feature Extraction
3. Classification

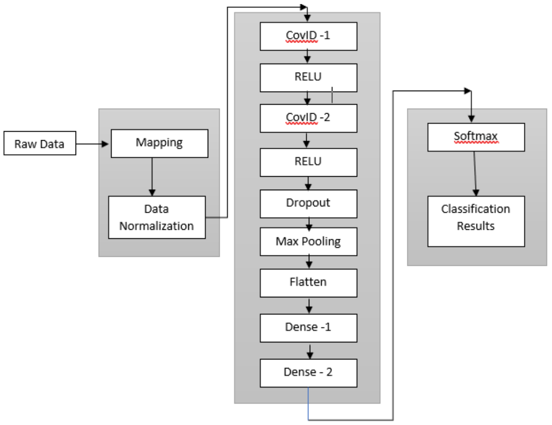


Fig. 1. Block diagram of the proposed model for intrusion detection

We use Cov1D layers which is 1D CNN layers wherein every Cov1D layer includes sixty-four convolution filters, forty features, three kernel sizes and for activating the layer we are using 'RELU' activation function to output the input directly if it is positive, otherwise, it's going to output zero. Next is a dropout layer which facilitates regularization of the output. Max pooling layer downsample or pool feature maps, which offers summarized version of the detected features in the input. Here we're using 1D max pooling. Flattening layer converts the output matrix into vectors, for smooth classification. At the final level of the module, we add one fully connected layer which includes two Dense layers. When we set 100 feature detectors and a 'Relu' activation function and 23 feature detectors and a "Softmax" activation function with hyperparameters, or combined model.

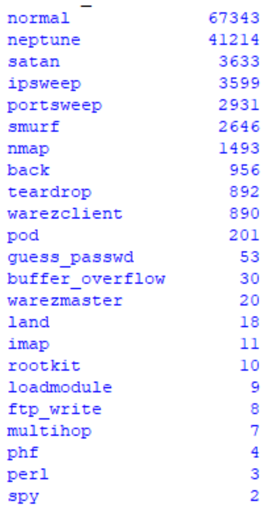
4. Experiments and Analysis

A. Dataset

The dataset used in this paper used for evaluation is the NSL-KDD dataset. It includes 125973 instances for training and 22544 instances for testing. The dataset consists of twenty-two forms of the latest attacks that are classified into four types:

- Denial of Service attack (DOS)
- User to Root attack (U2R)
- Remote to User attack (R2L)
- Probing

The NSL KDD dataset contains 42 features for each record out of which 38 are numerical features and remaining four are symbolic features which needed to be processed separately. Fig. 2 shows the normal and other 22 attacks included in the dataset and the number of samples available for each attack.



```
normal              67343
neptune             41214
satan                3633
ipsweep              3599
portsweep            2931
smurf                2646
nmap                 1493
back                  956
teardrop              892
warezclient           890
pod                   201
guess_passwd           53
buffer_overflow        30
warezmaster            20
land                   18
imap                   11
rootkit                10
loadmodule              9
ftp_write               8
multihop                7
phf                     4
perl                    3
spy                     2
```

Fig. 2. Different labels and no. of samples

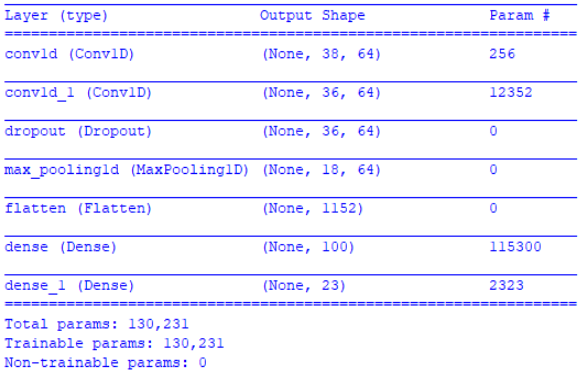
B. Data Preprocessing

The four symbolic features we go for digitalization, for example, the protocol-type feature contains three characters. i.e., TCP, UDP, ICMP, we convert it to [1, 0, 0], [0, 1, 0], [0, 0, 1], so we converted the 1D vector into a 3D vector. Similarly, the Label having twenty-three symbols which is converted to 23-dimensional vectors. For flag, which has eleven symbols, and gets transformed to 11- dimensional vectors. Lastly, the service feature that packs 70 symbols which gets transformed to a 70-dimensional vector. Thus, the Four symbolic features get mapped into 107-dimensional vectors.

The magnitudes of the numerical features could be highly varied because of the unmatching dimensions for numerical features. So, for eliminating the differences and normalizing the data Min-Maxscaler is used.

5. Experimental Evaluation

This experiment uses Accuracy (AC), Precision, Recall, and F1-score as an evaluation matrix.



```
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 38, 64)            256
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 36, 64)            12352
_________________________________________________________________
dropout (Dropout)            (None, 36, 64)            0
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 18, 64)            0
_________________________________________________________________
flatten (Flatten)            (None, 1152)              0
_________________________________________________________________
dense (Dense)                (None, 100)               115300
_________________________________________________________________
dense_1 (Dense)              (None, 23)                2323
=================================================================
Total params: 130,231
Trainable params: 130,231
Non-trainable params: 0
```

Fig. 3. The output shape of each layer

A. *Analysis of Experimental Results*

Our best-trained model contains two convolutional layers, one dropout layer, one max-pooling layer, one Flattening, and two dense layers. An input data of shape 40*1 is given to the convolutional layers. First, 1D convolutional layer give an output shape of 38*64 (64 denotes the number of filters), and second Convolutional layer constructs a 36*64 output shape. This is fed to the dropout layer which regularize the output of cov1D-2 layer. The Output of dropout is passed to the max-pooling layer with a pool size two. Pooling reduces the tensor shape to 18*64. A single column output of 1152 vectors transformed from the pooled feature map by flatten layer is given to the dense layers. Finally, two dense layers which help in classification is connected which gives an output shape of 23 vectors. This is the 23 different attacks.

For calculating the Loss function, categorical cross-entropy is used along with Adam optimizer. By changing the number of epochs, we can calculate the accuracy. From Fig. 4, it is clear that with a continuous increase in the number of the epoch, accuracy is rising. After 40 epochs the accuracy becomes constant which means the system goes overfitting which means max accuracy attained is 99.56% at 40 epochs with a loss of 1.96%, precision 76%, Recall 73%, and F1-score 74%.

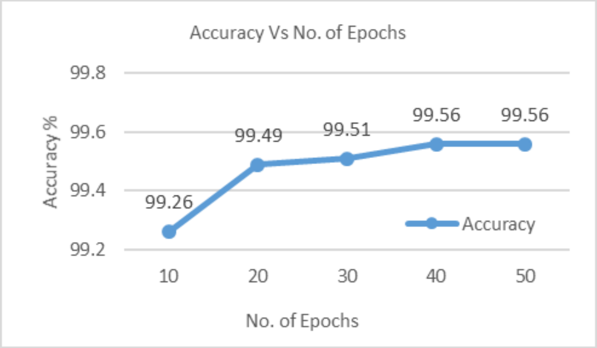


Fig. 4. Accuracy for different no. of epochs

Now SVM and Naïve Bayes based models are implemented for comparing it with our proposed model. Same 3 steps, i.e., data processing, feature extraction and classification in carried out for these to techniques also. Numerical values are processed for deleting unnecessary features and also symbolic features are converted to numerical values. In the case of SVM, we use SVC linear Classifier from sklearn library and for Naïve Bayes, we use GaussianNB from sklearn.naive_bayes. From the experiment, we can obtain Accuracy of 97.93%, Loss of 3.1%, Precision 58%, Recall 53%, F1-score 54% results, when we use the SVM algorithm. With Naïve Bayes, we could achieve relatively fewer results than SVM as shown in below Table 1, Accuracy 54.25%, Loss 46%, Precision 44%, Recall 60%, F1-score 42%.

Table 1
Comparison of CNN, SVM and Naïve Bayes

Model	**Accuracy%**	**Precision%**	**Recall %**	**F1-score %**
SVM	97.93	58	53	54
Naïve Bayes	54.25	44	60	42
Improved CNN	99.56	76	73	74

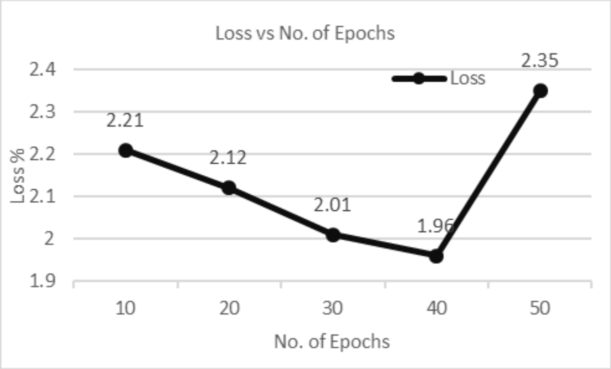


Fig. 5. Loss for different no. of epochs

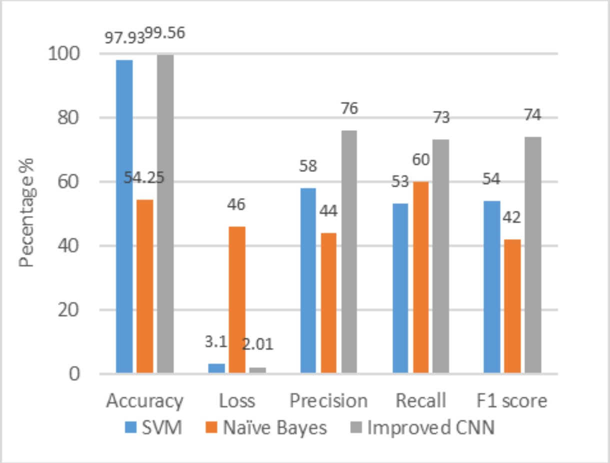


Fig. 6. Comparison of CNN, SVM and Naïve Bayes

6. Conclusion

A modified 1D-CNN based deep learning IDS is proposed in this paper. When compared with classification results of SVM and Naïve Bayes, we concluded that 1D-CNN outperforms other two traditional ML techniques for network intrusion detection. The proposed model obtained maximum accuracy of 99.56% for 40 epochs and observed to go overfitting when number of epochs is increased again. Fig. 6 shows the difference in Accuracy, Precision, recall, F1 score and Loss of three techniques that are compared in this paper. By analysing the graph, it is undoubtedly clear that 1D-CNN classifies the abnormal behaviour better when compared with ML techniques.

References

[1] R. U. Khan, X. Zhang, M. Alazab and R. Kumar, "An Improved Convolutional Neural Network Model for Intrusion Detection in Networks," *2019 Cybersecurity and Cyberforensics Conference (CCC)*, 2019, pp. 74-77.

[2] M. Azizjon, A. Jumabek and W. Kim, "1D CNN based network intrusion detection with normalization on imbalanced data*," 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2020, pp. 218-224\

[3] Anish Halimaa A, K. Sundarakantham, "Machine Learning Based Intrusion Detection Systems," in *Proceedings of the 3rd International Conference on Trends in Electronics and Informatics (ICOEI 2019)*.

[4] Setareh Roshan, Yoan Miche, Anton Akusok, Amaury Lendasse, "Adaptive and Online Network Intrusion Detection System using Clustering and Extreme Learning Machines," in *Journal of the Franklin Institute*, Volume.355, Issue 4, March 2018, pp. 1752-1779G.

[5] Iftikhar Ahmad, Mohammad Basheri, Muhammad Javed Iqbal, Aneel Raheem, "Performance Comparison of SVM, Random Forest, and ELM for Intrusion Detection," in *Survivability Strategies for Emerging Wireless Networks*, Volume. 6, May 2018, pp. 33789-33795.

[6] BuseGulAtli1, Yoan Miche, AapoKalliola, Ian Oliver, Silke Holtmanns, Amaury Lendasse, "Anomaly-Based Intrusion Detection Using Extreme Learning Machine and Aggregation of Network Traffic Statistics in Probability Space," in *Cognitive Computation*, June 2018, pp. 1-16.

[7] Pinjia He, Jieming Zhu, Shilin He, Jian Li, and Michael R. Lyu, "A Feature Reduced Intrusion Detection System Using ANN Classifier," in *Expert Systems with Applications*, Vol. 88, December 2017 pp. 249-247.

[8] Mazini, Mehrnaz, Babak Shirazi and Iraj Mahdavi, "Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms," in *Journal of King Saud University-Computer and Information Sciences*, 2018.

[9] Wathiq Laftah Al-Yaseen, Zulaiha Ali Othman, Mohd Zakree Ahmad Nazri, "Multi-Level Hybrid Support Vector Machine and Extreme Learning Machine Based on Modified K-means for Intrusion Detection System," in *Expert System with Applications,* Volume 66, Jan 2017, pp. 296-303.

[10] Vajiheh Hajisalem and Shahram Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection, in *Computer Networks*, vol. 37-50, pp. 37-50, 2018.

[11] Mohammed A. Ambusaidi, Xiangjian He, Priyadarsi Nanda, and Zhiyuan Tan, "Building an intrusion detection system using a filter-based feature selection algorithm," in *IEEE Transactions on Computers*, November 2014.

[12] Seyed Reza Hasani, Zulaiha Ali Othman and Seyed Mostafa Mousavi Kahaki, "Hybrid Feature Selection Algorithm for Intrusion Detection System," in *Journal of Computer Science,* 10 (6): 1015-1025, 2014.